1983

Personnel management database design for the Republic of Korea Air Force

Cha, Sang Ho

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PERSONNEL MANAGEMENT DATABASE DESIGN
FOR
THE REPUBLIC OF KOREA AIR FORCE

by

Sang Ho Cha

December 1983

Thesis Advisor: Daniel R. Dolk

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Personnel Management Database Design for the Republic of Korea Air Force		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Sang Ho Cha		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE December 1983
		13. NUMBER OF PAGES 97
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Database, Logical database, Database Management System, Schema Design, Decomposition, ORACLE

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this thesis we present a conceptual database for ROKAF's Personnel Management, based upon the relational data model. The objective of this thesis is to apply the computerized personnel information system to the area of military officer personnel management. A database design methodology which utilizes multiple levels of conceptual and logical database

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

design structure is presented. We discuss the logical schema design in terms of a stepwise, interactive process of specification and refinement. We present operations to manipulate the relational data model for end-users database processing during the integration process.

The logical database is intended for use within a relational database system. This model has been tested using ORACLE, a relational database management (DBMS) running on the VMS operating system of a VAX 11/780 computer.

Approved for public release; distribution unlimited.

Personnel Management Database Design
for
the Republic of Korea Air Force

by

Sang Ho Cha
Lieutenant Colonel, The Republic of Korea Air Force
B.S., The Korean Air Force Academy, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 1983

ABSTRACT

In this thesis we present a conceptual database for ROKAF's Personnel Management, based upon the relational data model. The objective of this thesis is to apply the computerized personnel information system to the area of military officer personnel management. A database design methodology which utilizes miltiple levels of conceptual and logical database design structure is presented. We discuss the logical schema design in terms of a stepwise, interactive process of specification and refinement. We present operations to manipulate the relational data model for end-users data base processing during the integration process.

The logical database is intended for use within a relational database system. This model has been tested using ORACLE, a relational database management (DBMS) running on the VMS operating system of a VAX 11/780 computer.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

A number of people assisted me during the writing of this thesis. First, I would like to express my deep gratitude to Professor Daniel R. Dolk of the Department of Administrative Science and Professors David K. Hsiao and Dusan A. Badal of the Department of Computer Science. In addition to giving encouragement, they read through the thesis, suggested improvements and corrections. I am also grateful to Mrs. Jeanne Bowers of the Department of Computer Science for laboratory support and guidance in the ORACLE implementation.

Finally, I must thank my wife Chun Hee for her understanding and encouragement throughout the work.

I. INTRODUCTION

Currently, the ROK Air Force Computer Center is developing a personnel data system with emphasis on collecting data and processing and presenting information to users in each staff department. But the collected data does not include all of the personnel data elements. Furthermore, not all of the collected data have been identified as data elements.

In order to strengthen the readiness of the ROK Air Force under the limited national defense expenditure, it is imperative that personnel management be performed very efficiently and that all data in connection with Combat Crew records and, in particular, for individual Pilot Quality Control (IPQC), be analyzed and integrated. To achieve this goal, the high level managers of Combat and Command (CAC) and Air Force Headquarters very often need a variety of data relevant to each personnel. This situation motivates the ROKAF to develop a modern database system.

A most important consideration in database development is to store data so that it can be used for a wide variety of applications and can be changed quickly and easily. In order to perform these functions, the data should be independent and functionally dependent on key values. It should also be possible to query the database to satisfy

user's requirements using application programs or the Database Management System (DBMS) itself. These data items should contain useful information for decision makers to analyze, plan and manage a personnel organization.

It is very difficult to develop a database which performs in an optimal fashion. There are many different ways in which data can be structured and each has its own advantages and disadvantages. Different users want to use different data. It is hardly possible to satisfy all of the users with one type of data organization. The normal form concepts of relational database models will be applied to develop a database for CAC and Air Force Headquarters personnel management. The relational data model supports data independence better than other models.

To use these databases for personnel management purposes, a commercially available database management and an end-user application system are needed. This thesis will therefore focus on a preliminary personnel relational database system. In chapter II, we discuss the general overview of a database system and a relational data model. In chapter III, we analyze the system requirements and develop the ROKAF personnel management database which includes all data and an end-user application system to extract useful data for the manager of CAC and Air Force headquarters. This chapter includes implementation of the developed database using ORACLE, a relational database

management system. Finally, in chapter IV, we present conclusions and recommendations based on the research presented in the thesis.

II. BACKGROUND

A. SYSTEMS ANALYSIS AND DESIGN

A system can be very simply and broadly defined as a group of interrelated or interacting elements. However, a more specific and appropriate concept of a system is utilized in data processing and computer technology. A system can be defined as a group of interrelated components that seeks the attainment of a common goal by accepting inputs and producing outputs in an organized transformation process.[Ref. 1]

Figure 2.1 illustrates the systems approach as a "recycling" process of systems development. This model summarizes the stages of the systems development cycle and can be applied to all systems. The "testing cycle" involves testing the model or system and performing any necessary redesign, reprogramming, and retesting activities. The "maintenance cycle" involves performing the systems development activities required to improve an established system.

1. Systems Analysis

Systems analysis involves analyzing in detail the information needs of prospective users and developing the system requirements of the proposed systems. The goal of systems analysis is to produce the system requirements of

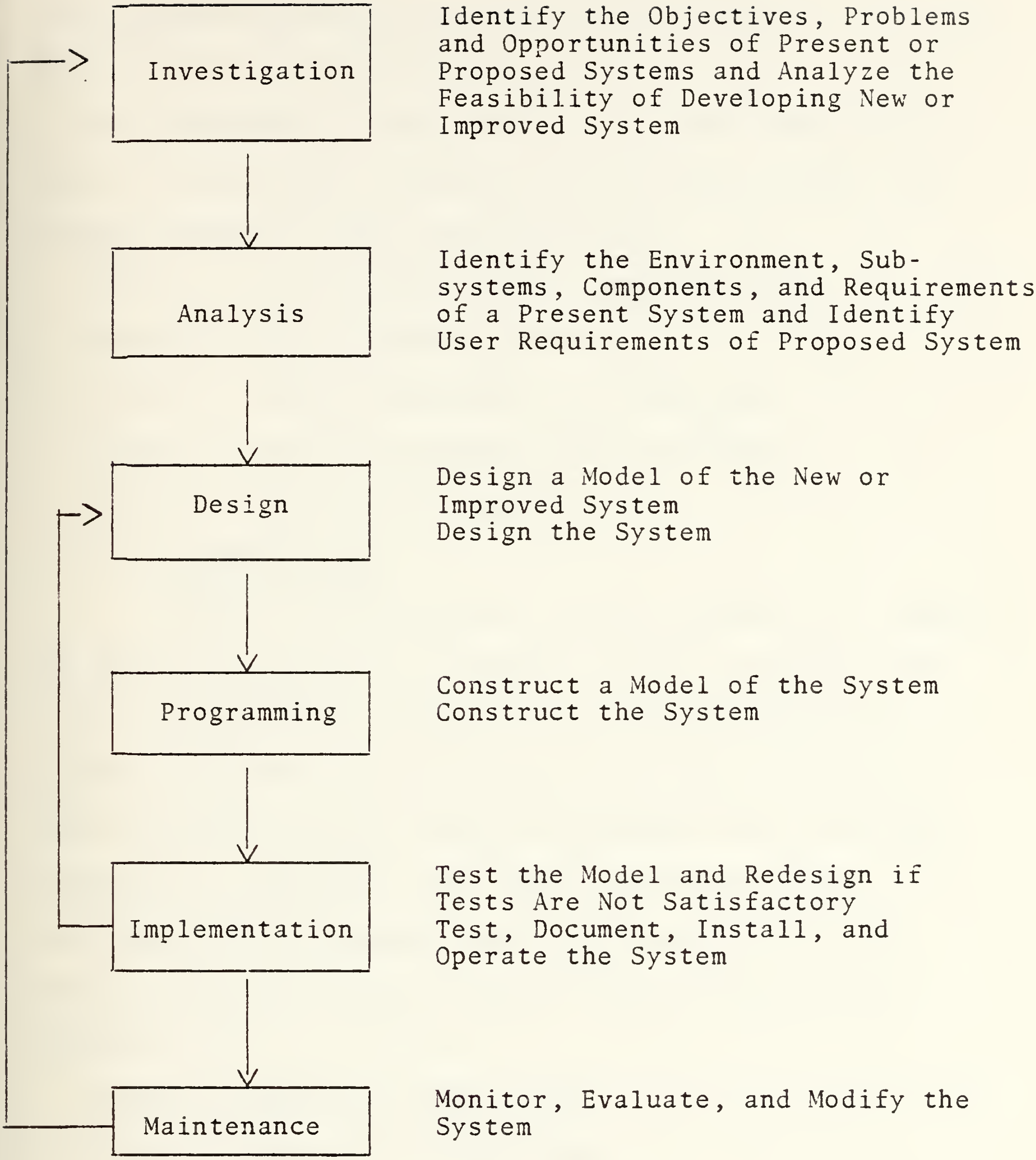


Figure 2.1 The Systems Development Cycle

the proposed database system. The system requirements describe the data processing and information requirements of

the proposed database system and are developed by a detailed analysis of (1) the organization that will use the system, (2) the information requirements of the user organization, and (3) the database system presently used, if any. The systems analysis stages are given below:

Phase 1 - organization system analysis

Phase 2 - major subsystem analysis

Phase 3 - present database system analysis

Phase 4 - proposed database system analysis

Phase 5 - system requirements

2. <u>Systems Design</u>

Systems design involves the development of a logical and physical design for a database system that meets the system requirements developed by the systems analysis process. Systems design involves the detailed design of input documents, output reports, database, and processing procedures. Personnel, data media, equipment, and programming specifications are also developed for the proposed system. Typical systems design steps are as follows:

Phase 1 - logical system design

Phase 2 - physical system design

Phase 3 - system specification

B. OVERVIEW OF A DATABASE SYSTEM

1. Introduction

Terminology for database systems is still not standardized. Different database systems employ different words to describe data and relationships.[Ref. 2] Therefore, confusion has arisen over the description of a database. However, it is to some extent accepted as conveying a more sophisticated concept than the older term "file." File processing systems are predecessors of database systems. They do not allow integrated processing.[Ref. 3] In order to develop a relational database system and to apply it, the general terminology and basic concepts must be understood by users and designers. This chapter covers the basic concepts of database system architecture, the relational data model, and database security.

2. Basic Concept of a Database System

a. Data versus Information

Data and information are meant to have two distinct meanings. Data refers to facts collected from observations or measurements, or to values physically recorded in a file or database. Information refers to the meaning assigned to those facts and values as a result of interpretation of data. Data are processed into information so that it can be understood and employed by users.[Ref. 1]

In some cases, data may not require processing before constituting information for a human user. However,

data are usually not useful until they have been subjected to a process where their form is manipulated and organized and their content is analyzed and evaluated. Then data become information. Figure 2.2 shows this relation.[Ref. 1]

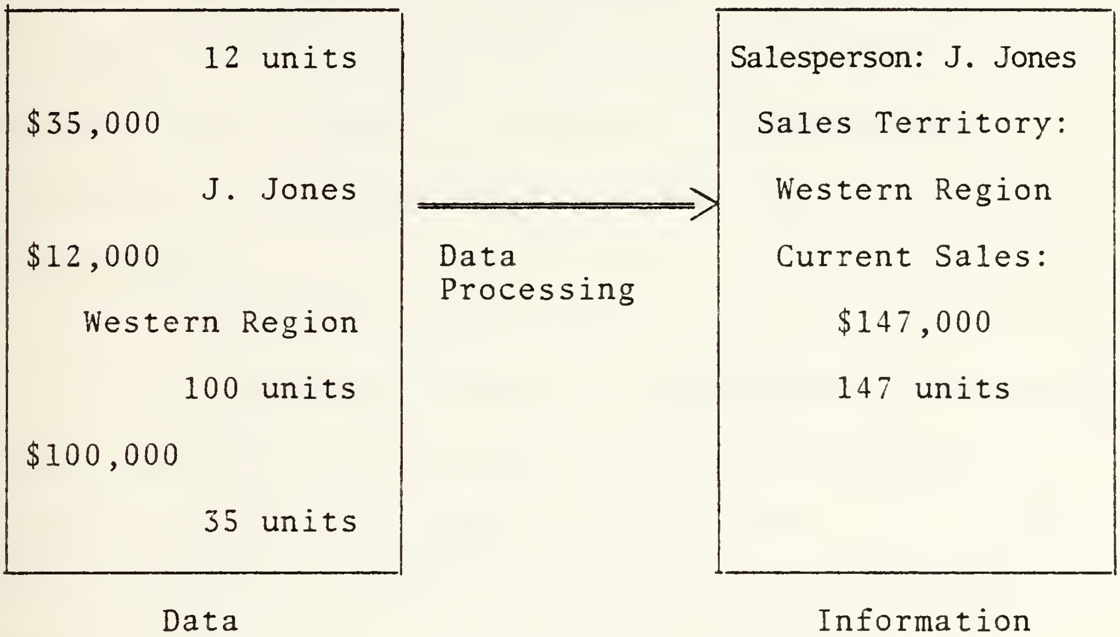


Figure 2.2 Data versus Information

b. Database

A database may be defined as a collection of data and data structure. There are two different databases: the physical database and the conceptual database.

The physical database resides permanently on secondary storage devices such as disks and tapes. We may view the physical database itself at several levels of abstraction, ranging from that of records and files in a programming language such as Pascal, through the level of

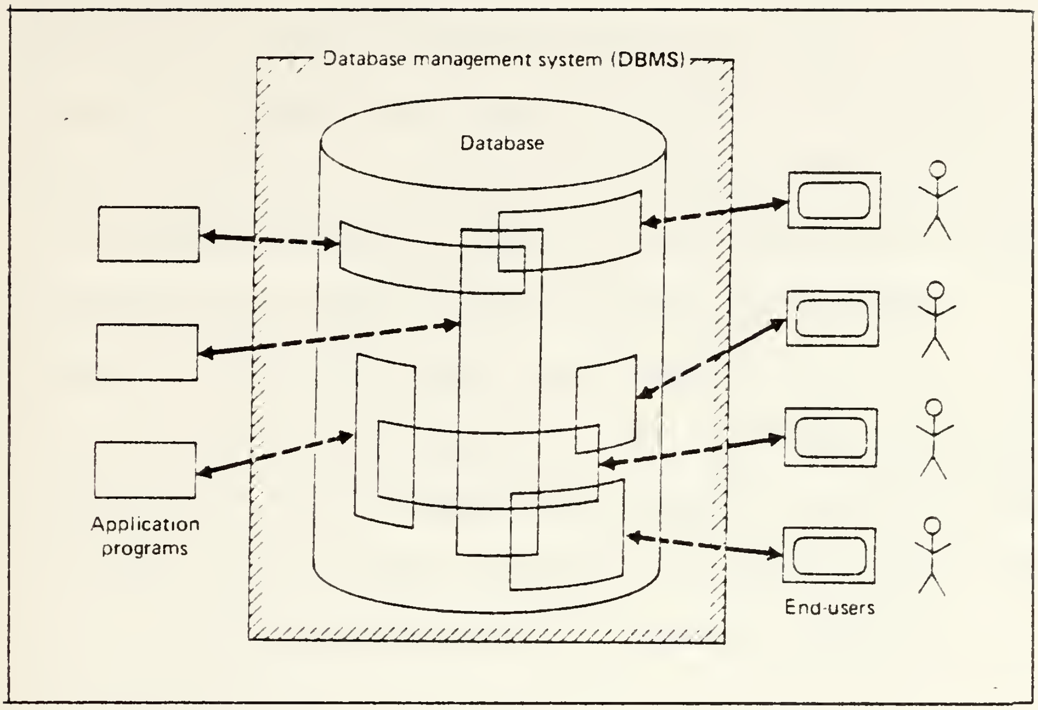
logical records as supported by the operating system underlying the DBMS, down to the level of bits and physical addresses on storage devices.

The conceptual database is an abstraction of the real world pertinent to an enterprise. A DBMS provides a data definition language to specify the conceptual schema and, most likely, some of the details regarding the transformation of the conceptual schema into the physical representation.

c. Database System

A database system is a combination of databases, a DBMS, and optionally an application system which is a collection of end-user application programs. C. J. Date [Ref. 4] defines a database system as a computer-based recordkeeping system; a system whose overall purpose is to record and maintain information. The information concerned can be anything that is deemed to be of significance to the organization. Figure 2.3 shows a greatly simplified view of a database system consisting of four major components: data, hardware, software, and users.

(1) Data. The data stored in the system is partitioned into one or more databases. A database system has several important advantages that accrue from having centralized control of the data. These are verified in Reference 4 as described below:



(Reprinted from [Ref. 4])

Figure 2.3 Simplified Database System

1. The data can be shared. This reduces the time needed to develop new systems or to respond to one-of-a-kind requests. In effect, more information can be obtained from existing data.
2. Redundancy can be reduced. This is the elimination or reduction of data duplication that can lead to a lack of data integrity in conflicting reports.
3. Inconsistency can be avoided. If a given fact is represented by a single entry, then inconsistency cannot occur.
4. Standards can be enforced. With central control of the database, the Database Administrator (DBA)

can ensure that all applicable standards are followed in the representation of the data.

5. Security restructions can be applied. The DBA can ensure that the only means of access to the database is through the proper channels, and hence can define authorization checks to be carried out whenever access to sensitive data is attempted.

6. Integrity can be maintained. The problem of ensuring that data in the database are accurate can be avoided by permitting the DBA to define validation procedures to be carried out whenever any update operation is attempted.

(2) Hardware. The hardware consists of the secondary storage volumes (disks and drums) on which the database resides, together with the associated devices, control units, channels, and so forth.

(3) Software. Between the physical database itself and the users of the system is a layer of software, usually called the Database Management System or DBMS.

A DBMS is an operating system for data that allows one or many persons to use and modify databases. A major role of the DBMS is to allow the user to deal with the data in abstract or logical terms, rather than as the computer stores the data. In this sense, the DBMS acts as an interpreter for a high-level programming language.

(4) Users. The users generally fall into three categories. First, there is the application programmer, responsible for writing application programs that use the database, typically in a language such as COBOL or PL/I.

The second class of user, is the end-user, accessing the database from a terminal. An end-user may employ a query language provided as an integral part of the system, or (s)he may invoke a user-written application program that accepts commands from the terminal and, in turn, issues requests to the DBMS on the end-user's behalf.

The third class of user is the database administrator, or DBA. A high-level person, generally called a DBA, is granted responsibility for matters that deal with the database as a whole, while individual queries and manipulations of the database are handled by the application programmers and users. The DBA's major responsibilities include determination of information content and access strategy, interfacing with users, performance monitoring, and defining crisis procedures for backup and recovery.

3. An Architecture for a Database System

The architecture is divided into three general levels: internal, conceptual and external.[Ref. 4] The internal level is the one closest to physical storage, that is, the one concerned with the way in which the data are actually stored. The external level is the one closest to users, that is, the one concerned with the way in which the

data are viewed by individual users. The conceptual level may be thought of as defining a community user view.

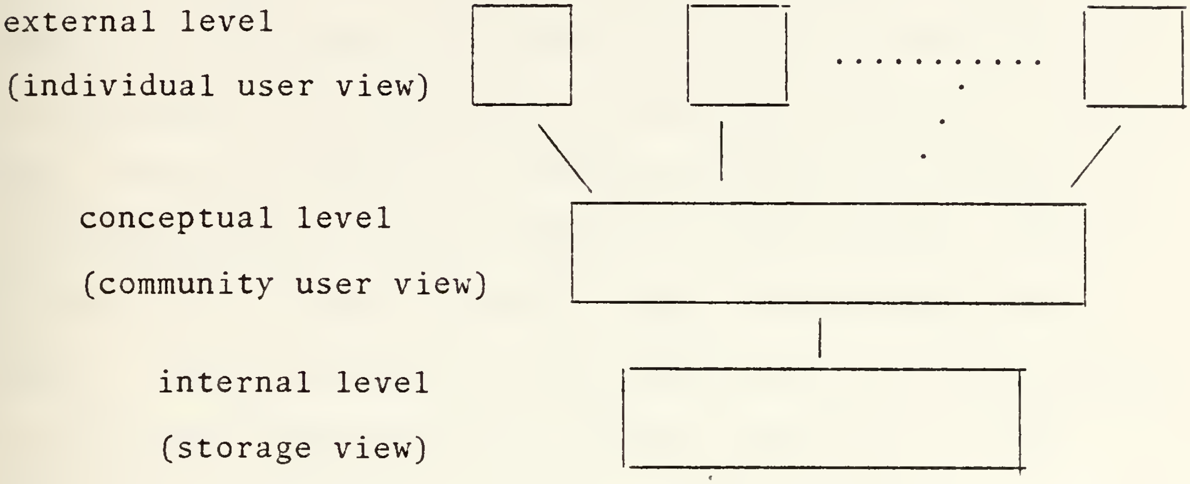


Figure 2.4 The Three Levels of Architecture

C. RELATIONAL DATA MODEL

1. Introduction

A data model based on relations and their representation as tables was first proposed by Codd.[Ref. 5] In the formulation of relational data models, the mathematical theory of relations is extended logically where required to meet data management objectives. The mathematical foundation of relational data models permits elegant and concise definition and deduction of their properties.[Ref. 6]

2. Structure

The only data structuring tool used by relational data models is a relation. The definition of a relation in

the relational data model is identical to the mathematical definition except that database relations are time varying. That is, tuples are inserted, deleted, and modified in database relations. The intension of a relational database is specified by a relational schema which consists of one or more relation schemes. A relation scheme is a listing of a relation name and its corresponding domain names, which can be used to represent an entity type in relational data models. Since relational data models are table data models, a relational schema does not reflect explicity all the relationships between relations in the database.

3. Constraints

A relation may have several keys. To eliminate the trivial keys, a key of relation must satisfy the following two properties.[Ref. 7]

1. Unique Identification. In each tuple of the relation the value of the key uniquely identifies each tuple, that is, no two rows have the same value for the attributes in the key taken as a whole.
2. Nonredundancy. No attributes that are part of the key can be removed without destroying property 1, that is, the key is minimal.

4. Query Language

Query languages for the relational model break down into two broad classes:

1. Algebraic languages where queries are expressed by applying specialized operators to relations, and
2. Predicate calculus languages, where queries describe a desired set of tuples by specifying a predicate the tuples must satisfy. The calculus-based languages have two classes, depending on whether the primitive objects are tuples or are elements of the domain of some attribute.

Each of the abstract query languages is equivalent in expressive power to the others and were first proposed by Codd [Ref. 8] to represent the minimum capability of any reasonable query language using the relational model.

Examples of the typical query languages are:

1. ISBL (Information System Base Language) is a query language developed at the IBM United Kingdom Scientific Center in Peterlee, England, for use in the experimental Peterlee Relational Test Vehicle (PRTV) system.
2. QUEL is the query language of INGRES, a relational DMBS developed at the University of California, Berkeley, to run under the UNIX operating system, a tuple calculus language.
3. Query-by-Example (QBE) is a language developed at IBM, a domain calculus language, Yorktown Hts.

D. DATABASE PROTECTION

1. Security

The subject of database security involves the protection of the database from unauthorized access. First, we need to protect against both undesired modification and/or destruction of data and against unauthorized reading of data. Three techniques are described below:

1. User Identification. The most common scheme to identify users is a password known only to the system and the individual.
2. Physical Protection. A high security system needs better identification that a password, such as personal recognition of the user by a guard.
3. Maintenance and Transmittal of Rights. The system needs to maintain a list of rights enjoyed by each user on each protected portion of the database.

2. Integrity Preservation

This aspect concerns nonmalicious errors and their prevention. The DBMS can help detect some programming bugs, such as a procedure that inserts a record with the same values in the key fields as a record that already exists in the database.

III. DATABASE AND APPLICATION SYSTEM DEVELOPMENT

A. USER REQUIREMENTS SPECIFICATION

1. Introduction

The ROK Air Force uses the general staff system of the USAF, namely, Personnel, Operations, Intelligence, Logistics, Plans and Operations.

The Air Force Headquarters has the responsibility for organization, training and equipping the ROK Air Force for the conduct of sustained combat operations. In national security the position of the ROK Air Force is critical because the Communist North Korean Air Bases are located very close to the capital city of Korea. North Korea is superior in numbers of aircraft, and stands face to face with Korea along the 155 mile Demilitarized Zone (DMZ). Another consideration is that the North Korean aircrafts are able to reach the capital city of the ROK within 3 or 4 minutes of crossing the DMZ line.

In order to strengthen the war potential of the Korean Air Force, it is imperative that personnel management for a small number of elite members under limited resources be performed very efficiently. The ROK government spends a rather large percentage of the total military budget for national defense, and the Department of National Defense spends a significant portion of the national defense

expenditure for personnel. The largest investment is in the ROK Air Force. In order to reduce the national defense expenditures and increase combat capabilities, the Air Force needs a computerized management information system for personnel management.

2. <u>Aspects of CAC and Air Force Headquarters Personnel Management</u>

Personnel managers need data about a specified individual's qualifications and a given unit's Average Level of Proficiency (ALP) in order to analyze, investigate, and plan for their organizations. Information about a specified individual's qualifications can be derived from functions involving procurement, education and training, assignment, treatment, promotion and retirement. Information about a unit's ALP can be derived from the collection of the individual personnel power data. It is important to increase individual and group proficiency in the personnel management field in order that the right people move into the right jobs at the right times and under the right circumstances. A specified individual's qualification becomes the basis for a given unit's ALP. Each factor of the individual's personnel management will be discussed based on Reference 9 and Reference 10.

3. <u>Personnel Administration</u>

a. Personnel Procurement

Personnel procurement is the process of gaining manpower for filling vacant positions which cannot be filled

from within the organization itself. Efficient personnel procurement requires information concerning the candidates that have been selected. Their relevant data must be kept and maintained so that they can be used at any time for transfer, new assignment, promotions, etc.

b. Personnel Education and Training

Information regarding the education and training of personnel is used mainly for personnel development and promotion. This information is used to match or minimize the difference between skills required to fill a position. A person's educational background can be used to gain special knowledge needed to place a person in a particular job and to prepare that person for a new assignment. Further, this information can be used to plan and monitor the careers of leaders, or those personnel with special abilities who will be future leaders, in order to extend their abilities and skills in preparation for future positions.

The results of personnel development can be measured by observing the performance of individuals in gaining necessary skills and abilities. This information can be recorded in the personnel database and used as a basis for further career development.

c. Personnel Assignment

Personnel assignment deals with selecting the right officers for the right positions. Three aspects must be considered for this job.

1. Every vacant position must be filled by a person with the ability to carry out the job in the best manner.
2. The capabilities and skills of each person must be fitted to the job so that he satisfies the job area.
3. Each person who is selected for a new position must have finished compulsory education and training courses, and must have carried out compulsory position in each rank.

d. Personnel Treatment

Personnel treatment deals with the physical and psychological aspects of person and job. These include such areas as mental and physical health, recreation, rewards, personnel service, transportation, salary, retirement plans, military insurance, annual pension and vacation (periodic, sick, reward, emergency). Mental and physical health conditions and rewards affect promotion and new assignment. Salary, military insurance, annual pension and personnel service affect the life of the family. Recreation, rewards, personnel service, transportation, retirement plans and vacations are very important for military morale.

e. Personnel Promotion

The promotion policy is that personnel who have finished minimum service duration in a rank and possess the capability to perform in upper level positions, be considered by a promotion selection committee. Therefore, the necessary information should be prepared and provided to

the decision makers, namely the promotion selection committee. The list of personnel who can be promoted should be provided according to rank and branch of service. The promotion point tables of all personnel should be provided by incorporating several items into these tables. These items are as follows:

1. The career which is required on current rank.
2. The result of efficiency reports which are taken annually on current rank.
3. Military education.
4. Rewards and punishment.
5. Physical and mental health condition, etc.

The promotion selection committee selects the officers to be promoted each year from officers who are recommended for promotion according to the above information. The necessary number of officers to be promoted each year are decided upon at the end of the previous year.

f. Personnel Separation

Personnel separation occurs when personnel voluntarily ask to be retired from the Air Force through the process of retirement or when someone cannot work or continue in the Air Force because of problems with their mental or physical health. Personnel who request retirement must have worked for the minimum public service duration in the Air Force. The minimum public service durations are different between resource organizations. If certain

individuals have attended a special service school, they must complete the additional term of obligation required for that school. However, if personnel reach the age limitation, rank limitation or maximum public service duration, they must retire on that day. Therefore, retirement information should be prepared and provided to decision makers (i.e., retirement selection committee). This information will include public service duration, a list of officers who wish to retire and have satisfied the minimum requirements, and a list of officers who can no longer work in the Air Force.

4. User Required Information for Personnel Management

The main functions of personnel management for the ROK Air Force have been described. Next, the information needed to analyze, investigate, plan, and apply those functions is described. Information which personnel managers may request might include:

1. A list of all new officers for each source organization including attributes such as academic ability, classification of home town, next of kin, health condition, completion rate of education and training, etc.
2. The number of cadets or candidates who should be inducted in the next year or at a specified year for each source organization.
3. List allocation of all officers by rank and by military education and training course.

4. Summary of an officer's career from a certain previous rank up to the current rank.
5. Total of classified flying time of all combat crews based on rank, qualified grade, capability grade, month and annual period, and organizational unit.
6. List of all retired combat crew, who can be recalled in the event of a national emergency, based on age and new occupation.
7. Present an information list for promotion purposes for each rank and branch, including career, result of efficiency record, education, rewards and punishments, health condition, and the order of promotion recommendation.
8. Present all information which is required for individual pilot quality control (IPQC).

All queries which may be made by personnel managers cannot be foreseen because different managers request different information. Personnal managers might need other information for their job in addition to that described above.

B. DATABASE DESIGN

1. Introduction

Database design is one of the most important steps in the development of a computerized information system. Size and complexity combine to make this task disproportionately time consuming and expensive.

Developing a database is an evolutionary process with the objective being an "idealized database." This is information that contains all the necessary data about all facets of an organization's operations and from which can be extracted instantaneously, in any form desired, information in response to inquiries in any format.

There are many ways in which a database can be designed. Generally, database design consists of two separate components: logical design and physical design. We will consider only a logical design technique for a relational database model.

2. Structure of a Relational Database Model

a. Relations

The data structuring tool used by the relational database model is a relation which is simply a two-dimensional table. Figure 3.1 illustrates a relation called PERSON, of degree 4. The four domains contain set values representing, respectively, RANK, Service Number (SN), Military Occupation Specialty (MOS), and NAME.

PERSON

RANK	SN	MOS	NAME
Major	50001	1124	Kim, Su Koo
Colonel	49345	1356	Cha, Sang Ho
Captain	58367	1523	Park, Ki Soo

Figure 3.1 An Example of A Relation

The RANK domain, for example, is the set of all valid ranks. Note that there may be ranks included in this domain that do not actually appear in the PERSON relation at this particular time. As the figure illustrates, it is convenient to represent a relation as a table. Each row of the table represents one n-tuble of the relation. The number of tuples in the PERSON relation is three.

b. Domains and Attributes

A domain is the set of possible values that an attribute can have. That is, "Cha, Sang Ho" is a value of attribute NAME. An attribute is the property of an entity which assocaites a value from a domain with each entity. For example, the relation PERSON is defined with four attributes (RANK, SN, MOS, NAME), and each attribute is drawn from a corresponding domain.

c. Keys

A key can be considered an attribute or a set of attributes which uniquely identify each entity in an entity set. For example, attribute SN of the PERSON relation has the property that each PERSON tuple contains a distinct SN value, and this value may be used to distinguish that tuple from all others in the relation. SN is said to be the primary key for PERSON.

3. <u>Schema Design</u>

A relational database is specified by a relational schema which consists of one or more relational subschemas.

A relational subschema is a listing of a relation name and its corresponding attributes. Figure 3.2 represents an example of a relational schema for ROK Air Force's personnel management.

```
PERSON (RANK, SN, MOS, NAME)
EDUCATION (SN, RANK AT THAT TIME, SCHOOL NAME,
   COUNTRY, LOCATION, PERIOD, MAJOR, RESULT GRADE)
```

Figure 3.2 An Example of a Relational Schema

There are four steps required in order to design a relation schema:[Ref. 11]

1. Determine the information requirements for the different areas of the organization involved in the schema design.
2. Express the information requirements as an enterprise description.
3. Obtain a database description which more rigorously defines the database structures and constraints, and satisfies the information requirements.
4. Check the schema for performance requirements of the prospective users.

The data are modeled initially and administered thereafter by people in certain roles.[Ref. 12] Those people can be classified as follows:[Ref. 13]

1. The enterprise administrator specifies the enterprise description (conceptual schema).
2. The database administrator is concerned with specifying the physical aspects of the database description (internal schema).
3. The application administrators provide the multiple views (external schemas) for the various application areas within an organization.

Each administrator is responsible for providing a particular view of the necessary data, the relevant relationships among the data, and the rules and mappings between views. Each administrator role uses tolls and techniques as provided by data models for the successful description and operation of the database.

a. Requirements Analysis

The first step of schema design is requirements analysis. This step consists of a high-level analysis of the function of an organization. The functions of the departments of personnel management given in the previous part of this chapter are an example of requirements analysis. The purpose of this step is to:

1. Gain familiarity with the area of the organization to be modeled.
2. Determine the information requirements of the organization without regard to constraints other than the way in which the organization does business.

3. Represent these requirements via some formal modeling technique.

The main purpose of requirements analysis is to understand the user's needs. Subsequent steps of the schema design process can transform these needs to subschemas according to the relational data model.

In order to meet the requirements, there are two views: How the organization operates and what is required to support the operation. The how and what are aspects of an organization which can be represented in terms of the functions of the organization and the data classes that support these functions.

(1) Function Classes. A function in an organization is an essential activity or decision required to manage the resources and operations of the organization. Functions in an organization are identified by:

1. Examining statements of purpose of a task or an organizational area.
2. Examining work programs in an organizational area.
3. Identifying products or services provided by an organizational area and determining what functions are needed to produce such products and services.

Following the above principles, we have defined the functions of the Department of the ROK Air Force Personnel Management. The functions for Headquarters are procurement, education and training, assignment, treatment,

promotion, and separation policies. In particular, the functions for CAC are assignment, education and training, and IPQC policies for combat crews only.

(2) Data Classes. Before discussing data classes, we extract specific data groupings from the present file system in different functional areas. These files, relevant to each person, consist of data elements which are required by different users in different functional areas. These files are: Person, Military Training, Salary, Medical Records, Inspector, and Supply Record. Furthermore, these files are independent of one another, therefore, all data are not shared by all organization areas.

A trend toward integrated file structures has resulted in the grouping of all data elements relevant to the management and operations section of a user organization. The emerging database concept requires placing all relevant data in one database in a consistent and standardized manner, eliminating unnecessary duplication and file handling, and providing selective inquiry and extraction capabilities designed to meet a wide variety of information requests. Therefore, data classes must be well organized in order to achieve the goals of this system.

A data class in an organization is an aggregation of data (attributes) that is required by a function or is produced by it. Data classes in an organization are identified by examining the data required

or produced by a function. The functions and corresponding data classes for personnel management are shown in Figure 3.3.

(3) Data Dictionary. The generation of the data dictionary which documents functions, data classes, and their interrelationship should be initiated at this point.[Ref. 14]

Individual DBMS have their own methods for defining data descriptions. Each has a repository for the data base description, a language facility to process that description, and a mechanism to input that description to the DBMS. A comprehensive dictionary will include cross-reference information such as which programs use which pieces of data, which departments require which reports, and so on. The general objectives of a data dictionary are to provide:[Ref. 15]

1. Various reporting facilities such as cross-reference reports, changes effecting reports, error-reports, etc.
2. Various retrieval capabilities such as keywording, indexing, and online or batch querying.
3. Common language to control, retrieve and update the data dictionary.
4. Validation and redundancy-checking capabilities.
5. Security safeguards to control access to the data dictionary.
6. Data description generation.

Area	Function	Area	Function
1	Procurement	4	Promotion
2	Education and Training	5	Separation
3	Assignment	6	Treatment

DATA CLASS	FUNCTION					
	1	2	3	4	5	6
PERSON	*	*	*	*	*	*
IMMEDIATE FAMILY	*					*
LINEAL FAMILY	*					*
MILITARY TRAINING		*				
MILITARY CAREER		*	*	*		
EDUCATION		*	*	*		
FOREIGN LANGUAGE PROFICIENCY	*	*	*			
HEALTH CONDITION			*	*	*	*
AWARD/PUNISHMENT			*	*	*	*
EFFICIENCY RECORD		*	*	*		
COMBAT TRAINING/QUALITY		*	*			
REQUIRED SERVICE	*	*	*	*	*	*
PROMOTION LIST		*	*	*	*	
ASSIGNMENT POLICY	*	*	*	*	*	*
TEMPORARY DUTY			*			
PERSONNEL SUPPLY		*			*	*
SALARY						*
VACATION						*

Figure 3.3 Functions and Data Classes

The data dictionary contains all the information from logical data description to the detailed physical data description. For instance, Figure 3.4 shows the schema of the relation PERSON as it might appear in a data dictionary.

b. Enterprise Description

Enterprise description consists of five phases. The first phase is identifying all the entities of interest to each organizational area, the relationships between them, and any constraints which may exist. The first phase results in a view of the schema for each organizational area. These views are then integrated to form an enterprise description which describes the entire conceptual schema. This description is used mainly for communication between the users and the schema designers. For each entity type identified, a description of the entity type is produced and the associated data classes identified. The description names the entity type, defines what it represents, and lists its associated attributes. Entity type identification is an iterative process. The description of an entity type may change many times before everyone agrees that it is right. The entity types identified for personnel management are shown in Appendix A Part III.

In the second phase, the relationship between entities are identified from the functions. In order to do this, there are several considerations to be taken into account:

PERSON	(RANK, SN, MOS, NAME)
RANK	: present rank of the officer
SN	: service number
MOS	: number of military occupational specialty
Name	: name of the officer

Figure 3.4 An Example of PERSON Schema Data Dictionary Entry

1. For each function, what are the known correspondences between entity types associated with the function?
2. What is the appropriate name for each relationship type?
3. What is the meaning of each relationship type, either formally or informally?
4. What combinations of relationship types make sense as separate, identifiable relationship types?

The relationship among entity sets is simple an ordered list of entity sets. A particular entity set may appear more than once on the list. The relationship types obtained from this process are shown in Figure 3.5.

PROMOTION -- function
PERSON EFFICIENCY -- between PERSON and EFFICIENCY REPORT
PERSON CAREER -- between PERSON and MILITARY CAREER

Figure 3.5 An Example of Relationship Types

The next phase is to complete the enterprise description step, by identifying constraints on the attributes, entity types, and relationship types. It seems

better to state all constraints explicitly rather than as inherent constraints. To help identify constraints, the following questions are posed:

1. What is the domain of values for each attribute?
2. What are the known functional dependencies between attributes of each entity type?
3. What are the keys for each entity type?
4. What is the mapping property of each relationship?
5. What are the predicate constraints to be placed upon the data?

Functional dependencies will be discussed in detail in the next section.

It is difficult to arrive at a set of constraints that represents the application and its consistent and feasible, because some forms of the constraints are difficult to understand and are prone to misunderstandings and errors. The result of this phase of the enterprise description step is a list of the entity types and their attributes. The results of this step are also identified in Appendix A Parts I and III.

The fourth phase of the enterprise description step integrates views for each organizational area into one enterprise description. The enterprise description is a synthesis of the information requirements of each organizational area. Documentation of the enterprise description consists of summarizing the data obtained from

the interviews in a suitable manner. It also includes retention of the universe of discourse on who uses each entity type and relationship type. It may be necessary to iterate by negotiating with each organizational area until all organizational areas agree that the enterprise description accurately reflects their information requirements.

The final phase of the enterprise description step identifies the transaction-processing requirements of the organization with respect to the enterprise description. All current and projected transactions are included. For each transaction, the designer identifies its nature (retrieval, update, delete, insert), its frequency, its origin (organizational area), and its purpose, together with the point(s) of the schema it affects. The previous four steps are used as a basis for describing the transactions. To help identify requirements for supporting transactions, the following questions are posed:

1. What transactions are required by each organizational area?
2. What entity types, attributes, and relationship types are involved in each transaction?
3. What is a sketchy outline of each transaction in terms of the enterprise description in English or a problem specification language?
4. What kind of access is required by each transaction?

5. What is the mode of operation of each transaction?
6. What is the frequency of each transaction?
7. What is the processing priority of each transaction?
8. What is the need for concurrent update activity?
9. What kind of pattern of database usage do we expect?
10. What reports are needed?
11. What is the format of each report?
12. What is the acceptable time frame for producing each report?
13. What security requirements are important?

The result of this step is a list of all transactions and their characteristics. Figure 3.6 contains a simple example of transactions required for personnel management. The list of transactions is shown to the different organizational areas and an agreement on a final list is reached together with some priorities for implementation. The overall results of this final step are contained in Appendix A Part IV.

c. Database Description

This step transforms the enterprise description into a database description which means a description of the proposed schema according to the data model in the target DBMS. We will illustrate the process for the relational approach.

For a relational DBMS, the enterprise description is transformed into a relational schema

(Figure 3.7). Entities are mapped into base relations which are permanently stored in the database. Relationship types are mapped into base relations if they are information bearing. Non-information-bearing relationship types can be mapped into derived relations (i.e., joins).

Transaction: List of all officers who have excellent ability in the German language, hold the rank of Captain, whose MOS is 2214, have a Master's degree and whom have graduated from the Air Force Academy.

Entity: PERSON, FOREIGN LANGUAGE PROFICIENCY (FLP), EDUCATION

Relationship types: PERSON FLP, PERSON EDUCATION

1. Retrieve PERSON entity (for RANK, MOS, COMMISSION TYPE)
2. Retrieve all PERSON entities related to the FLP via a PERSON FLP.
3. Retrieve all PERSON entities related to the EDUCATION via PERSON EDUCATION.

Figure 3.6 A Simple Example of Transaction

The result of transforming the enterprise description into a database description represents a documentation of the schema. In addition, we have a sketch of each transaction to be performed. The schemas and transaction sketches should again be discussed with the different organizational areas in order to obtain each organization's approval.

PERSON (SN, RANK, NAME, RRN, MOS, PERMANENT ADDRESS
PRESENT ADDRESS, COMMISSION TYPE, COMMISSION
DATE, BRANCH OF SERVICE, MARITAL STATUS,
RELIGION)
key: SN

FLP (SN, TYPE OF LANGUAGE, PROFICIENCY DEGREE)
key: SN

ASSIGNMENT POLICY (UNIT NAME, RANK, REQUIRED DUTY,
MOS, PREREQUISITE POSITION, PREREQUISITE
EDUCATION)
key: UNIT NAME

REQUIRED SERVICE (SN, FINAL YEAR OF TOTAL REQUIRED
SERVICE DURATION, MAXIMUM AGE FOR REQUIRED
SERVICE AT THAT RANK, MAXIMUM DURATION AT THAT
RANK, FINAL YEAR OF MAXIMUM DURATION AT THAT
RANK)
key: SN

PERSON FLP (SN, RANK, NAME, TYPE OF LANGUAGE)
key: SN

ASSIGNMENT POLICY REQUIRED SERVICE (UNIT NAME, RANK,
FINAL YEAR OF TOTAL REQUIRED SERVICE DURATION)
key: UNIT NAME

Figure 3.7 A Relational Schema for Procurement Policy

4. Schema Analysis

The major direction of the design effort is to obtain an accurate schema, that is a schema representing the

database application on which the transactions of the application can be serviced. Given an abstract schema, what are the desired properties it should have and how does one transform the schema into another equivalent schema with the desired properties?

Basically, a schema consists of structure and constraints. Constraints can be used as a guideline for deciding the schema's structure according to three criteria: representation, nonredundancy, and separation. Representation should be a guideline for getting a good schema. Non-redundency states that a constraint that can be derived from the structures and other constraints already specified in a schema, should not be redundantly specified. Separation requires that we structure the schema in such a way that information units, as represented by constraints, are separated.

Constraints can be used as a yardstick to evaluate and manipulate schemas. The most well understood and simple type of constraint deals with dependencies between attributes in a schema.

There is a tremendous amount of choice in schema design regarding structuring of the data and the specification of constraints. Many different schemas can be associated with the same application. It would be nice to come up with one that is "good" and "right." "Good" usually means a schema that provides reasonable database

performance. Database performance is a function of physical database design, but it is not covered in this thesis. "Right" usually means that the schema reflects the real properties of the world the designers are trying to represent. This latter point is discussed in the following section.

a. Functional Dependencies

The functional notion of functional dependence (FD) can be defined as: Given a relation R, attribute Y of R is functionally dependent on attribute X of R if and only if each X-value in R has associated with it precisely one Y-value in R (at any one time). That is, if f: X -> Y, then Y is said to be functionally dependent on X, and X is said to functionally determine Y. When there is only one functional dependency f from X to Y, the notion X -> Y is used as an abbreviation.

In the Person-and-Family database, for example, attribute RANK, NAME, and MOS of relation PERSON are each functionally dependent on attribute SN, because, given a particular value for SN, there exist precisely one corresponding value for each of RANK, NAME, and MOS.

The other type of FD can be defined as: Given a relation R, attribute Y of R is functionally dependent on attribute X of R if and only if, whenever two tuples of R agree on this X-value, they also agree on their Y-value. For example, relation PERSON in Figure 3.8 satisfies the FD

RANK -> MOS. Furthermore, the attribute BASIC SALARY of relation SALARY is functionally dependent on the composite attribute (RANK, SALARY STEP).

PERSON

RANK	SN	NAME	MOS
Captain	11330	CHA, Sang Ho	1121
Major	12220	Kim, Ho Soo	1023
Captain	13110	Chung, Ko Ja	1121

SALARY

RANK	SN	SALARY STEP	BASIC SALARY	F-ALLOWANCE
Captain	11330	5	2300	100
Major	12220	3	2200	200
Captain	13110	2	2000	150

FAMILY

SN	SPOUSE NAME	SPOUSE RRN	NO OF DEPENDENT
11330	Yoon, Sun Ja	11111-25364	2
12220	Park, Min Ok	12111-24561	4
13110	Shin, Mun So	13241-45326	3

Figure 3.8 The Person-and-Family Database: Relational View

We represent the FDs in an example set of relations by means of a functional dependency diagram. An example is shown in Figure 3.9.

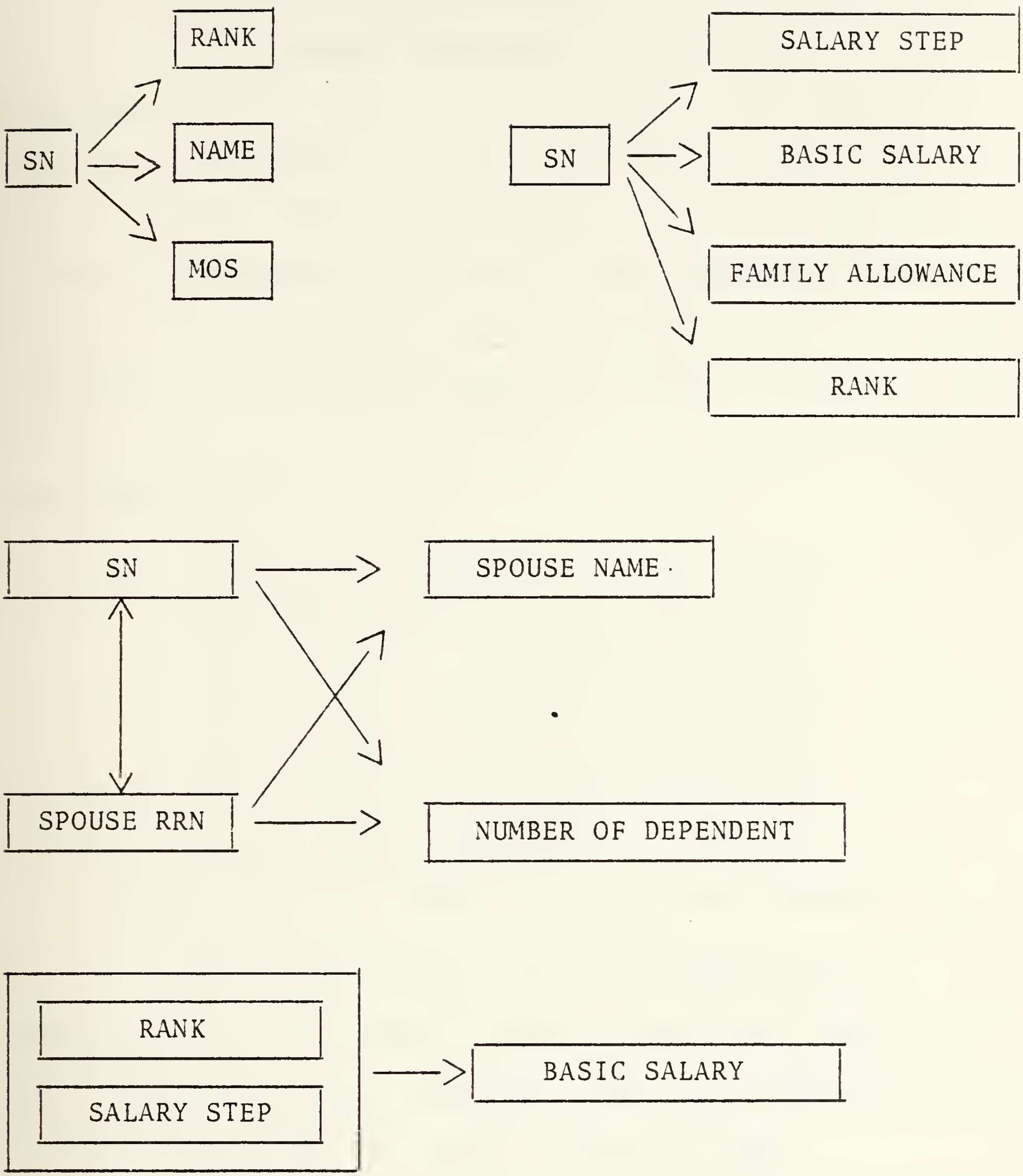


Figure 3.9 Functional Dependencies in Relation PERSON, FAMILY, and SALARY

b. Multivalued Dependencies

Multivalued dependencies (MVDs) are a generalization of functional dependencies. The idea of functional dependency is limited in the following sense. When A multidetermines B (A ->->B), A not only determines B, but it determines B uniquely. The uniqueness limitations can be too restrictive in some cases. Consider, for instance, the following example of a relation shown in Figure 3.10.

ASSIGNMENT POLICY

UNIT NAME	RANK	MOS
1100	CAPTAIN	1111
1100	MAJOR	1120
1120	CAPTAIN	1121
1100	COLONEL	1122

Figure 3.10 An Example of Multivalued Dependency

In this example, there is a multivalued dependency from UNIT NAME to RANK written UNIT NAME ->-> RANK. The multivalued dependency says that all RANKs depend on the UNIT NAME and not on the individual MOSs.

Functional and multivalued dependencies can be used to specify particular types of constraints on entity types. These constraints relate to properties of the mappings between the attributes of an entity type. These

properties are specified during schema design and can be used in the analysis of a schema to eliminate undesirable properties.

c. Decomposition of Relation Schemes

The decomposition of a relation scheme $R = (A_1, A_2, \ldots, A_n)$ is its replacement by a collection $\rho = (R_1, R_2, \ldots, R_n)$ of subsets of R such that $R = R_1 \cup R_2 \cup \ldots \cup R_n$. There is no requirement that the R_i's be disjoint.

Some peculiar conditions arise when we lump attributes together which should be kept apart. These conditions are called anomalies. We will illustrate these anomalies using the following relation scheme.

COMPANY (Employee#, Department#, Manager, Contract Type)

The following anomalies may arise in manipulating this relation.

1. Update anomaly. The change of a manager in a department necessitates a series of changes of this manager for each employee and contract type in which the department is involved. That is, a change must ripple through and cause a series of changes for the database to be consistent.
2. Insertion anomaly. When the first employee is hired for a department, a manager and contract type must be specified.

3. Deletion anomaly. When the last employee is fired, any department information will cease to exist. This can be considered an anomaly if we want to retain important, long-ranging information about the department.
4. Redundancy. The contract type and the manager of a department are repeated in many tuples. The above situation can be avoided by decomposition.

In the above example, the anomalies can be eliminated by breaking the relation scheme into two relational schemes.

Employee (Employee#, Department#)

Department (Department#, Manager, Contract Type)

In the decomposed schema, employees and departments are isolated and related only by specifying the department in which an employee works. Decomposition is based on the two functional dependencies Department# -> Manager, Contract Type and Employee# -> Department. The decomposition isolates these two dependencies in separate relation schemes. As a result, they do not interfere with each other. These side effects are eliminated by isolating the dependencies in different relation schemes.

A decomposition is considered "good" when the schema ρ is equivalent to R and when it eliminates some of the anomalies. For the two schemas R and ρ to be equivalent, the following two properties are necessary:

1. Lossless join. More formally, the join of relations R_1 and R_2 is lossless if $R_1 \cup R_2 \rightarrow R_1$ or $R_1 \cup R_2 \rightarrow R_2$. Also, it can be applied to multivalued dependencies.
2. Dependency preservation. A decomposition is dependency preserving if its dependencies are preserved within the new relation schema.

d. Normal Forms for Relation Schemes

Normalization theory is built around the concept of normal forms. A relation is said to be in a particular normal form if it satisfies a certain specified set of constraints. For example, a relation is said to be in first normal form if and only if it satisfies the constraint that it contains atomic values only. Codd originally defined first, second, and third normal forms in Reference 7. That is, all normalized relations are in 1NF; some 1NF relations are also in 2NF; and some 2NF relations are also in 3NF. The motivation behind the definitions is that 2NF is "more desirable" than 1NF, in a sense to be explained, and similarly 3NF is more desirable than 2NF.

We have chosen 3NF relations in designing the database described in the previous section, rather than 2NF or 1NF relations or other more intricate normal forms (4NF, for example) which are very difficult to implement. We will discuss only 1NF, 2NF, and 3NF here. The way that a relation scheme is turned into a normal form relation scheme is by decomposition.

A relation is in first normal form if every attribute is a simple attribute. That is, there are no composite attributes. For instance, consider the relation scheme.

SALARY (RANK, SALARY STEP, BASIC SALARY, ALLOWANCE (FAMILY, MOS), TAX)

The attribute allowance is obviously composite. We can eliminate this by constructing a new relation scheme as follows:

SALARY (RANK, SALARY STEP, BASIC SALARY, FAMILY ALLOWANCE, MOS ALLOWANCE, TAX)

This example suggests a general algorithm for putting a relation scheme into 1NF. We just expand the relation scheme by eliminating all composite attributes and replacing them with their constituent parts.

For this particular relation the key is RANK, SALARY STEP. However, a SALARY relation that is already in 1NF is said to be in 2NF since it has no partial dependencies of nonprime attributes on keys. If a partial dependency exists, it will result in update, insertion and deletion anomalies.

A relation is third normal if it is in 2NF and it has no transitive dependencies of nonprime attributes on keys. In Figure 3.11 SALARY1 is in 2NF but has the

transitive dependency RANK, SALARY STEP -> BASIC SALARY -> TAX, where TAX -/-> RANK, SALARY STEP and BASIC SALARY -/-> RANK, SALARY STEP. This transitive dependency results in update, insertion, and deletion anomalies for values of the tax attribute as well as redundency of tax values. To get rid of this transitive dependency, the relation SALARY1 can be decomposed into two relation schemes and extensions shown in Figure 3.12.

SALARY1

RANK	SALARY STEP	BASIC SALARY	TAX
WO	9	1000	90
WO	10	1050	95
Captain	3	1000	90
Captain	4	1050	95
Major	1	1200	100

RANK, SALARY STEP -> BASIC SALARY

BASIC SALARY -> TAX

Figure 3.11 Example Relation in 1NF and 2NF

A relation is in 3NF if and only if, for all time, each tuple of R consists of a primary key value that identifies some entity, together with a set of mutually

independent attribute values that describe that entity in some way. In particular, Appendix A Part III is based on 3NF.

SALARY2

RANK	SALARY STEP	BASIC SALARY
WO	9	1000
WO	10	1050
Captain	3	1000
Captain	4	1050
Major	1	1200

TAXATION

BASIC SALARY	TAX
1000	90
1050	95
1200	100

Figure 3.12 Example Relations in 3NF

C. IMPLEMENTATION OF A PROTOTYPE DATABASE USING ORACLE

1. Introduction

The ORACLE relational database management system is a computer program that manages pieces of information stored in a computer. ORACLE offers the ease of use, functionality, and flexibility of a modern relational DBMS without compromising system performance. While users perceive their

data as tables, ORACLE employs sophisticated data storage and access techniques to optimize system performance. Today, ORACLE is being used in large production applications and online transactions involving databases with billions of characters of data.

ORACLE uses the SQL (Structured Query Language) data language. SQL is a query language, a data manipulation language, a data definition language, and a data control language. With conventional systems, we have to learn a different language for each one of those functions. All SQL facilities can be used directly from a terminal or embedded in programming languages such as COBOL, FORTRAN, BASIC, and PLI. All users - regardless of who they are or what they are doing - use the same language, SQL.

In this section, we will implement the data manipulation aspects of ORACLE. In particular, we present the DML portions of the SQL language. The SQL DML operates on both base tables and views. All examples will be based on the PERSON-and-SALARY database of Appendix A Part III. The results of examples of the selected operation are appended to Appendix B Part I.

2. SQL Description

a. Tuple Relational Calculus

The concept of a relational calculus specifically tailored to a relational database was first proposed by Codd.[Ref. 8] There are the two forms of

relational calculus, called tuple relational calculus and domain relational calculus. A tuple calculus expression is essentially a nonprocedural definition of some relation in terms of some given set of relations. Formally, the expressions of the tuple calculus are from the following elements:

1. Each tuple variable is constrained to range over some named relation. If tuple variable T represents tuple t, then the expression T.A represents the A-component of t, where A is an attribute of the relation over which T ranges.
2. Conditions of the form X * Y, where * is any one of =, /=, , =, , or =, and at least one of X and Y is an expression of the form T.A and the other is either a similar expression or a constant.

b. Basic SQL Commands

(1) <u>Creating a Table</u>. We create a table using the CREATE TABLE command:

```
CREATE TABLE PERSON (SN number(6), RANK char(10), NAME
char(10), BIRTHDATE char(10), MOS number(4), CD char(10),
CT number(3), MS number(3))
```

(2) <u>Inserting Data Into a Table</u>. Immediately after a table is created, rows can be entered into the table using the insert command. The following command was used to enter the first row into the PERSON table.

```
INSERT INTO PERSON VALUES(111111, 'MAJOR','CHA, S. H.',
'08-MAR-42', 1111, '10-MAR-70', 1, 1)
```

In the insert command we name the table(PERSON) into which the row is to be inserted and list data values that go into each column.

(3) Other SQL Commands. There are other SQL commands for other functions:

1. SELECT - This command has at least two part: a) The SELECT clause lists the column we want to retrieve. b) The FROM clause names the tables from which to retrieve the columns.
2. DELETE - Remove a row from a table.
3. UPDATE - Modify a field in a row.

c. Retrieval Operations

The fundamental operation in SQL is the mapping, represented syntactically as a SELECT-FROM-WHERE block. A WHERE clause causes a "search" to be made, and only tuples that meet the search condition are retrieved. A WHERE clause search condition can use any of the following comparison operators; =, !=, , =, , and =.

For example, the query "Get officer's RANK and NAME whose MOS is 1111," may be expressed as follows:

```
SELECT      RANK, NAME
FROM        PERSON
WHERE       MOS = 1111
```

RESULT

RANK	NAME
Major Colonel	CHA, S. H. KIM, D. J.

From this example we can see that the "mapping" operation is effectively a horizontal subsetting followed by a vertical subsetting. Results of a wide variety of retrieval queries are displayed in Appendix B Part II.

d. Group Functions Operations

In all the examples so far, we have selected values stored in each row of a table or values calculated for each row. That is, we have selected information about individual rows stored in database. We can also select "summary" information about groups of rows in the database.

ORACLE provides five group-functions that can be applied to data retrieved in a query:

1. AVG - Complete the average value
2. SUM - Computes the total value
3. MIN - Finds the minimum value
4. MAX - Finds the maximum value
5. Count - Counts the number of values

3. <u>Data Security</u>

ORACLE allows users to share access to the same database. While users can share data if desired, ORACLE will automatically keep data private unless a user explicitly gives another user access to it.

1. ORACLE requires users are to provide user USERID and PASSWORD when logging on.
2. Sharing data with other users. When a user creates a table, the user become the owner of that table. Only the owner can use the table unless he wants to share it with other ORACLE users. Users can give and take away access to their tables with the SQL command:

GRANT - give other users access to user's tables.

REVOKE - take away other users access to user's tables.

The GRANT command is made up of three basic clauses:

GRANT - a function (SELECT, INSERT, UPDATE, ALTER, INDEX, CLUSTER)

ON - a table or view

TO - a user

For example: Have user CHA grant the SELECT privilege on the PERSON table to a user named KIM.

```
GRANT   SELECT
ON      PERSON
TO      KIM
```

Once a privilege has been granted it may be withdrawn by means of the REVOKE command. Privileges are revoked from the named grantee and from all users to whom he has granted them. For example: Revoke from KIM the right to INSERT into the PERSON table.

```
REVOKE  INSERT
ON      PERSON
FROM    KIM
```

3. The database administrator. When the ORACLE database is first created, there is only one user authorized to log on. That user has the name SYSTEM and a password MANAGER. If a user logs on under the name SYSTEM, he has full DBA authority and can create other users. Example: Create a new user named CHA with DBA authority.

```
CONNECT SYSTEM/MANAGER
GRANT DBA TO KIM IDENTIFIED BY CHA
Example: Log on as KIM
CONNECT KIM/CHA
```

4. Changing user password. A user with DBA authority (like KIM or SYSTEM) can change another user's password. For example, user could change the password of SYSTEM to something other than MANAGER. This will prevent other users logging on with the user name SYSTEM and gaining DBA access to the database.

As mentioned in the above ORACLE DBMS, we find facilities to prevent incorrect data from being in a database and to prevent the reading of data that should not be disclosed to unauthorized personnel in the ROKAF's environment.

4. Data Dictionary

The ORACLE data dictionary is a group of tables that contain information about the database. These dictionary tables are created by ORACLE at the time the database is created. The dictionary describes tables, indexes, clusters, users, access privileges and other things in the database. ORACLE automatically updates the dictionary whenever anyone creates or drops a table or view and grants or revokes a privilege, so the dictionary always contains a current description of the database.

Users can read dictionary tables using standard SQL queries and, since the dictionary is "self describing," it can be queried to determine the names of its own tables, columns, etc. However, not all of the dictionary tables and views have the SELECT privilege granted to the public. Other tables, such as the one containing user passwords, are only accessible to someone with DBA privileges. The results of examples of selected dictionary queries are displayed in Appendix B Part III.

5. Report Generation

In the personnel database, all data in relation to personnel management are a variety of format types and prompt actions. The development of new personnel management techniques and the continuously changing requirements of each organization may require new formats or alternations of existing formats. If we do not need formal report

documentation, then the staff and management of each organization unit can directly use the terminal display. Otherwise, we can use ORACLE report facilities which provide 5 commands for producing formatted reports:

1. Column - Format a column's heading and data
2. TTITLE - Put a title on the top of the page
3. BTITLE - Put a title on the bottom of the page
4. Break - Break the report up into groups of rows
5. Compute - Compute subtotals and totals

The output of any SQL query can be automatically formatted into a report with page and column headings and page numbers. Column headings and data formats are taken from the data dictionary. Users can request tables on control breaks, alter report and column headings, and specify page size and column justification. The results of examples of selected queries are shown in Appendix B Part IV.

IV. CONCLUSIONS

This thesis has focused on the Korean Air Force Headquarters and Command and Control, however its findings are applicable to all departments of the Korean military. The developed database presented here is based on a relational database model and a computerized personnel management system for military officers only, however it may very well form the basis of the total personnel management system.

This thesis examined a stepwise design process for logical design of personnel database which covers:

1. Collecting data relevant to personnel and analyzing each organization's requirements.
2. Identifying all the entity and relationship types, and constraints for each organizational area.
3. Using normal forms based upon functional dependencies for eliminating anomalies and redundancy.
4. Designing the database dictionary which communicates between database designers and users.

The data dictionary was designed as a directory for the data and is included in Appendix A.

Implementation of a prototype database using ORACLE resulted in a more effective and timely presentation of all required personnel information. This DBMS is particularly

appropriate to the application programmers and end-users who are working in the ROKAF who do not have much experience with database systems. This database system can increase personnel management's efficiency and decrease staff work load as well as reduce national defense expenditure.

This database can serve as a prototype from which ROKAF can refine existing schemas and develop further applications such as report generation in the Korean language.

APPENDIX A

DATA DICTIONARY

I. LIST OF DATA ELEMENTS AND DOMAINS

The letter A corresponds to alphanumeric, while the letter N corresponds to numeric. Each format number represents the maximum number of digits for a value range.

ATTRIBUTE	FORMAT	REMARK
AAP	N1	assignment to a position
AG-E	N3	average grade of efficiency point
BOS	N2	branch of service
BS	N8	basic salary
CD	N8	commission date
CEC	N1	commission education course
C-GRADE	N1	capability grade
COUNTRY	N2	name of country
CS	N2	class standing
CT	N1	commission type
DATE	N4	year month day
DC-ADVANCE	N8	date of capability advance
DEGREE	N1	graduate degree
D-NAME	A20	name of daughter
D-OCCUP	N2	daughter's occupation
DPP	N1	data point for promotion

ATTRIBUTE	FORMAT	REMARK
DPR	N8	date of promotion
DQ-ADVANCE	N8	date of quality advance
D-RRN	N15	RRN of daughter
DUTIES	N3	duties
ED	N8	evaluated date of efficiency
ESD	N8	date of estimated supply
E-SN	N6	service number of evaluator
E-YEAR	N4	examination year
FA	N7	family allowance
FMDR	N4	final year of maximum duration at that rank
F-NAME	A20	name of father
F-OCCUP	N2	father's occupation
F-PM-AD	A40	father's PM-AD
F-RRN	N15	RRN of father
FRS	N4	final year of total required service duration
G-CREDIT	N4	final credit
HEIGHT	N3	height
IB	N7	intelligence benefit
LOC	A20	city, state of indication country
LSA	N7	long service allowance
MAJOR	N2	major course of education school
MD	N8	marriage date
MDR	N2	maximum duration at that rank

ATTRIBUTE	FORMAT	REMARK
M-NAME	A20	name of mother
M-OCCUP	N2	mother's occupation
MOS	N4	military occupational specialty
MOSA	N7	MOS allowance
M-RRN	N15	RRN of mother
MRS	N2	maximum age for required service at that rank
MS	N1	marriage status
MTC-NAME	N3	name of military training course
NAME	A20	name of person
NOF	N1	number of family
OAG	N3	order of average grade
OCCUP	N2	type of occupation
OOP	N3	order or promotion
PD	N1	proficiency degree
PE	N2	prerequisite education
PERIOD	N16	year month day (19801009-19831009)
PM-AD	A40	permanent address
PP	N2	prerequisite position
PR-AD	A40	present address
Q-GRADE	N1	quality grade
RANK	N2	rank of person
RB	N6	regular bonus
R-DUTIES	N3	required duties
REASON	A50	type of reason

ATTRIBUTE	FORMAT	REMARK
RE-DATE	N8	regular examination date
RESULT	N1	result status of training
RRN	N15	resident registration number
SCHOOL NAME	A10	name of school
SD	N8	supplied date
SEX	N1	sex
SI	N3	supply items
SN	N6	service number
S-NAME	A20	name of spouse
S-OCCUP	N2	spouse's occupation
SON NAME	A20	name of son
SON OCCUP	N2	son's occupation
SON-RRN	N15	RRN of son
S-PERIOD	N2	total serviced year
S-PM-AD	A40	spouse's PM-AD
S-RANK	N1	status at that rank
S-RRN	N15	RRN of spouse
SS	N2	salary step
TAX	N6	tax
TOAD	A10	type of award/punishment
TOB	N1	blood type
TOE	N3	total number of evaluatee
TOL	N1	type of language
TOP	N1	type of promotion

ATTRIBUTE	FORMAT	REMARK
TOPM	A10	type of punishment
TOV	N1	type of vacation
TRS	N1	type of required service
TTDY	N2	type of temporary duty
UNIT NAME	N4	name of military unit
WEIGHT	N3	weight

II. LIST OF CODE NUMBER FOR DATA ELEMENTS

Each of these tables contains two elements: code and description. For instance, "01 General" indicates code number is General.

1. RANK/S-RANK:

   01 General　　02 Lieutenant General

   03 Major General　　04 Brigadier General

   05 Colonel　　06 Lieutenant Colonel

   07 Major　　08 Captain

   09 Second Lieutenant　　10 First Lieutenant

   11 Warrant Officer

2. SEX:

   1 Male　　2 Female

3. MILITARY OCCUPATIONAL SPECIALTY:

   1111 supply　　2222 intelligence

   (not included fully here for security reasons)

4. COMMISSION SOURCE:

1 AF Academy	2 ROTC

(not included fully here for security reasons)

5. MARRIAGE STATUS:

1 Married	2 Single

6. TYPE OF RELIGION:

1 Buddhist	2 Catholic
3 Protestant	4 Other

7. UNIT NAME:

2211 2211th Training Wing

1122 1122th Supply Company

(not included fully here for security reasons)

8. NAME OF MILITARY TRAINING COURSE:

111 Escape and Evasion

222 Sea Survival

(not included fully here for security reasons)

9. COUNTRY:

01 USA	02 JAPAN
03 ENGLAND	04 FRANCE
05 INDONESIA	06 GREECE
07 PHILLIPINES	08 CHINA
09 GERMAN	10 CANADA

10. DEGREE:

1 PH.D	2 Master's Degree
3 Bachelor's Degree	

11. QUALITY GRADE:

1 Low Quality	2 Medium Quality
3 High Quality	4 Superior Quality

12. CAPABILITY GRADE:

1 Low Capability	2 Medium Capability
3 High Capability	4 Superior Capability

13. COMMISSION EDUCATION COURSE:

1 AF Academy	2 Reserve officers' training

14. TYPE OF REQUIRED SERVICE:

1 Long Service	2 Short Service

15. SUPPLY ITEMS:

01 Combat Shoe	02 Hat
03 Gloves	04 Pants
05 Raincoat	06 Necktie

(not included fully here for military security reasons)

16. TYPE OF VACATION:

1 Regular vacation	2 Convalescent Leave
3 Reward vacation	4 Emergency Leave

17. TYPE OF LANGUAGE:

1 English	2 Japanese
3 French	4 Chinese
5 German	

18. BLOOD TYPE:

1 A	2 B
3 O	4 AB

19. TYPE OF PROMOTION:

1 Regular	2 Meritorious

20. TYPE OF FLIGHT:

(not included here for military security reasons)

21. TRAINING REQUIREMENT:

(not included here for military security reasons)

22. RESULT STATUS OF TRAINING:

1 graduate	2 not graduate

23. OCCUPATION:

01 farming	02 national public service personnel
03 serviceman	04 education public service
05 commerce	06 fisheries
07 student	08 industry

24. ASSIGNMENT TO A POSITION/DUTIES/REQUIRED DUTIES:

XXX company command

(not included fully here for security reasons)

III. THE RELATIONAL SCHEMA OF DATABASE DOMAIN

See part II of data dictionary.

RELATION

1. PERSON (SN, RANK, NAME, RRN, MOS, PM-AD, PR-AD, CT, C D , MS, BOS, RELIGION)

   Primary Key: SN

2. IMMEDIATE FAMILY (SN, MD, S-RRN, S-NAME, S-PM-AD, S-OCCUP, SON-RRN, SON-NAME, SON-OCCUP, D-RRN, D-NAME, D-OCCUP)

   Primary Key: SN + MD

3. LINEAR FAMILY (SN, F-RRN, F-NAME, F-OCCUP, M-RRN, M-NAME, M-OCCUP, B-RRN, B-NAME, B-OCCUP, S-RRN, S-NAME, S-OCCUP)

   Primary Key: SN

4. MILITARY TRAINING (SN, S-RANK, CS, RESULT)

   Primary Key: SN

5. MILITARY TRAINING COURSE (MTC-NAME, PERIOD, UNIT NAME)

   Primary Key: MTC-NAME

   This subschema can be manipulated from 4 using DBMS.

6. MILITARY CAREER (SN, UNIT NAME, S-RANK, APP, PERIOD)

   Primary Key: SN

7. EDUCATION (SN, S-RANK, PERIOD, MAJOR, DEGREE, G-GRADE, DPP)

   Primary Key: SN

8. EDUCATION COURSE (SCHOOL NAME, COUNTRY, LOC)

   Primary Key: SCHOOL NAME + COUNTRY

9. HEALTH CONDITION RECORD (SN, E-YEAR, RE-DATE, SEX, HEIGHT, WEIGHT, TOB)

   Primary Key: SN + E-YEAR

10. AWARD/PUNISHMENT (SN, S-RANK, TOAP, DATE, DPP, REASON)

    Primary Key: SN + TOAP + DATE

11. AWARDED LIST (SN, NAME, DUTIES)

    Primary Key: SN + NAME

    This subschema can be manipulated from 10 using DBMS.

12. PROMOTION POINT (TOAP, DPP)

Primary Key: TOAP

13. EFFICIENCY RECORD (SN, ED, AG-E, TOE, OAG, E-SN)

Primary Key: SN + ED

14. COMBAT QUALITY/TRAINING (SN, MOS, UNIT NAME, Q-GRADE, C-GRADE, DQ-ADVANCE, DC-ADVANCE)

Primary Key: SN + MOS

15. REQUISITE ASSIGNMENT RECORD

(not included here for military security reasons)

This subschema can be manipulated from 14 using DBMS.

16. PROMOTION LIST (SN, S-RANK, TOP, DPR, OOP)

Primary Key: SN

17. ASSIGNMENT POLICY (UNIT NAME, RANK, R-DUTIES, MOS, PP, PE)

Primary Key: MOS + R-DUTIES

18. REQUIRED SERVICE (SN, FRS, MRS, MDR, FMDR)

Primary Key: SN

19. COMMISSION LIST (CEC, TRS)

Primary Key: CEC

20. TEMPORARY DUTY (SN, TTDY, PERIOD, LOC, COUNTRY)

Primary Key: SN

21. SALARY (SN, IB, TAX, RB)

Primary Key: SN

22. BASIC SALARY (RANK, SS, BS)

Primary Key: RANK + SS

This subschema can be manipulated from 21 using DBMS.

23. FAMILY ALLOWANCE (NOF, FA)

Primary Key: NOF

This subschema can be manipulated from 21 using DBMS.

24. LONG SERVICE PERIOD (S-PERIOD, LSA)

Primary Key: S-PERIOD

This subschema can be manipulated from 23 using DBMS.

25. MOS ALLOWANCE (MOS, MOSA)

Primary Key: MOS

This subschema can be manipulated from 21 using DBMS.

26. VACATION LIST (SN, TOV, PERIOD)

Primary Key: SN + TOV

27. FOREIGN LANGUAGE PROFICIENCY (SN, TOL, PD)

Primary Key: SN, TOL

28. PERSONNEL SUPPLY (SN, SI, SD, ESD)

Primary Key: SN, SI

29. SUPPLY ITEMS

(not included here for security reasons)

IV. A SAMPLE LIST OF PROJECTED TRANSACTION

These samples of transaction-processing are very helpful to end user application system designers. For each transaction, we identify its nature (retrieval, update, insert, delete), its frequency, its organization, and its purpose, together with the part of the schema it affects.

As following in enterprise description in section B, we identify requirement for supporting transaction. Some of the relevant description are correspond to each number order.

1. organizational area
2. entity types and attributes
3. relationship type
4. type of access (retrievel, update, delete, insert)
5. mode of operation (batch, online)
6. frequency (daily, weekly, monthly, yearly, required day)
7. processing priority (I, II, III, VI)
8. security requirement (I, II, III)
9. report format

1. A list of all new officers who are service number, rank, RRN, final education school name, major course, degree, commission type, and whose MOS is 1111.

1. operation department
2. PERSON (SN, RANK, MOS, RRN, CT)
   EDUCATION (SN, MAJOR, DEGREE)
   EDUCATION COURSE (SCHOOL NAME)
3. PERSON-EDUCATION
   EDUCATION-EDUCATIONCOURSE
4. retrieval
5. batch
6. yearly
7. IV
8. I
9.

RANK	SN	NAME	RRN	MOS	S-NAME	MAJOR	DEGREE	CT

2. List of all combat crews who have some classified qualification, capability grade, and whose rank are captain.

1. operational department
2. PERSON (SN, RANK, NAME, MOS)

   COMBAT QUALITY/TRAINING (SN, MOS, UNIT NAME, Q-GRADE, C-GRADE)
3. PERSON-COMBAT QUALITY/TRAINING
4. retrieval
5. batch
6. required day
7. I
8. I
9.

SN	NAME	UNIT NAME

3. Average of classified flying of all combat crew based on certain rank, quality and capability grade, annual period, and organizational unit.

1. operational department
2. PERSON (SN, RANK)

   COMBAT QUALITY/TRAINING (SN, UNIT NAME, Q-GRADE, C-GRADE)

   REQUIREMENT FULFILLNESS RECORD (TYPE OF FLYING, FLYING TIME)
3. PERSON-COMBAT QUALITY/TRAINING

   COMBAT QUALITY-REQUIREMENT FULFILLNESS RECORD

4. retrieval

5. batch

6. monthly

7. I

8. I

9.

UNIT NAME	Q-GRADE	C-GRADE	TOF	AVERAGE TIME

4. Update list of all retired person whose MOS was 111 including present address, age.

1. personnel department

2. PERSON (SN, NAME, PRESENT ADDRESS, RETIRE STATUS, MOS, RRN)

3. PERSON

4. update

5. batch

6. monthly

7. III

8. III

9.

SN	NAME	PRESENT-ADDRESS	RRN	AGE

APPENDIX B

SAMPLE QUERIES OF ORACLE IMPLEMENTATION

I. TABLE OPERATION

A. CREATING TABLES

1. Create a table named PERSON.

```
UFI> CREATE TABLE PERSON(SN NUMBER(6),RANK CHAR(10),NAME CHAR(10),
  2  MOS NUMBER(5),CD CHAR(10),CT NUMBER(3),MS NUMBER(3));

Table created.
```

2. Insert three data into table named PERSON.

```
UFI> INSERT INTO PERSON VALUES(234567,'COLONEL','CHA,S.H.',111,
  2  '11-MAR-65',1,1);

1 record created.

UFI> INSERT INTO PERSON VALUES(245678,'MAJOR','KIM,K.S.',121,
  2  '20-APR-66',1,1);

1 record created.

UFI> INSERT INTO PERSON VALUES(245001,'MAJOR','PARK,S.U.',130,
  2  '21-SEP-66',1,2);

1 record created.
```

3. List the PERSON table.

```
UFI> SELECT     * FROM PERSON;

        SN RANK       NAME              MOS CD                CT         MS
---------- ---------- ---------- ---------- ---------- ---------- ----------
    234567 COLONEL    CHA,S.H.          111 11-MAR-65          1          1
    245678 MAJOR      KIM,K.S.          121 20-APR-66          1          1
    245001 MAJOR      PARK,S.U.         130 21-SEP-66          1          2
```

B. MANIPULATION OF TABLE

1. Adding a new column to an existing table.

Q1: List all officer's current salary

```
UFI> SELECT     * FROM SALARY;
```

SN	SSTEP	IB	RB	BS
234567	2	200	1000	960
245678	5	170	880	820
245011	2	170	800	710
290000	2		500	480
256000	5	960	700	650
214111	4	200	1200	1100
214000	5	200	1250	1200
290123	1		450	410

8 records selected

Q2: Add a column TAX to the SALARY table.

```
UFI> ALTER TABLE SALARY
  2  ADD         (TAX NUMBER);

Table altered
```

Q3: List the SALARY table.

```
UFI> SELECT     * FROM SALARY;
```

SN	SSTEP	IB	RB	BS	TAX
234567	2	200	1000	960	
245678	5	170	880	820	
245011	2	170	800	710	
290000	2		500	480	
256000	5	960	700	650	
214111	4	200	1200	1100	
214000	5	200	1250	1200	
290123	1		450	410	

8 records selected.

2. Updating rows in a table

Q4: List service number 234567's current salary.

```
UFI> SELECT     * FROM SALARY WHERE SN = 234567;
```

SN	SSTEP	IB	RB	BS	TAX
234567	2	200	1234	960	

Q5: Set service number 234567's regular bonus (RB) to 1400.

```
UFI> UPDATE SALARY
  2  SET         RB = 1400
  3  WHERE SN = 234567;

1 record updated.
```

Q6: Verify that service number 234567's Regular Bonus (RB) has been updated.

```
UFI> SELECT      * FROM SALARY WHERE SN = 234567;
```

SN	SSTEP	IB	RB	BS	TAX
234567	2	200	1400	960	

3. Deleting rows from a table.

Q7: List the SALARY table.

```
UFI> SELECT      * FROM SALARY;
```

SN	SSTEP	IB	RB	BS	TAX
234567	2	200	1234	960	
245678	5	170	880	820	
245011	2	170	800	710	
290000	2		500	480	
256000	5	960	700	650	
214111	4	200	1200	1100	
214000	5	200	1250	1200	
290123	1		450	410	

```
8 records selected.
```

Q8: Delete service number 234567's row from the SALARY table.

```
UFI> DELETE FROM SALARY WHERE SN = 234567;

1 record deleted.
```

Q9: List the SALARY table.

```
UFI> SELECT      * FROM SALARY;
```

SN	SSTEP	IB	RB	BS	TAX
245678	5	170	880	820	
245011	2	170	800	710	
290000	2		500	480	
256000	5	960	700	650	
214111	4	200	1200	1100	
214000	5	200	1250	1200	
290123	1		450	410	

7 records selecte

II. RETRIEVAL OPERATION

A. GENERAL OPERATION

1. Selecting data from a Table.

Q10: Select SN, RANK, NAME, and MOS of all officers from the PERSON table.

```
UFI> SELECT SN,RANK,NAME,MOS FROM PERSON;
```

SN	RANK	NAME	MOS
234567	COLONEL	CHA,S.H.	111
245678	MAJOR	KIM,K.S.	121
245011	MAJOR	PARK,S.U.	130
290000	LIEUTENANT	LIM,S.N.	111
214111	COLONEL	JANG,U.I.	130
290123	LIEUTENANT	UI,C.H.	222
256000	CAPTAIN	CHU,K.S.	111
214000	COLONEL	YOON,I.S.	121

2. Selecting Specific Rows from a Table.

Q11: Select only all officers whose MOS is 111.

```
UFI> SELECT      * FROM PERSON WHERE MOS = 111;
```

SN	RANK	NAME	MOS	CD	CT	MS
290000	LIEUTENANT	LIM,S.N.	111	16-MAY-80	2	2
256000	CAPTAIN	CHU,K.S.	111	11-FEB-67	1	2
234567	COLONEL	CHA,S.H.	111	11-MAR-65	1	1

Q12: Find SN, RANK, and NAME of all officers with regular bonus (RB) greater than $1000.

```
UFI> SELECT SN,SSTEP,IB,RB FROM SALARY WHERE RB > 1000;

        SN      SSTEP         IB         RB
---------- ---------- ---------- ----------
    214111          4        200       1200
    214000          5        200       1250
```

3. Selecting Rows that satisfy Multiple Conditions.

Q13: List SN, RANK, and NAME of all officers whose MOS is 111 and MS is 1.

```
UFI> SELECT SN,RANK,NAME FROM PERSON WHERE MOS = 111 AND MS = 1;

        SN RANK       NAME
---------- ---------- ----------
    234567 COLONEL    CHA,S.H.
```

4. Selecting Rows within a certain range.

Q14: Find SN, RANK, NAME, and RB of all officers whose regular bonus is between $700 and $900.

```
UFI> SELECT PERSON.SN,RANK,NAME,RB FROM SALARY,PERSON
  2  WHERE PERSON.SN = SALARY.SN
  3  AND RB BETWEEN 700 AND 900;

        SN RANK       NAME               RB
---------- ---------- ---------- ----------
    245678 MAJOR      KIM,K.S.          880
    256000 CAPTAIN    CHU,K.S.          700
```

5. Null in Search Condition.

Q15: List SN, RANK, NAME, IB, BS, and RB of all officers who do not receive IB.

```
UFI> SELECT PERSON.SN, RANK, NAME,IB,RB,BS FROM SALARY,PERSON
  2  WHERE PERSON.SN = SALARY.SN
  3  AND IB IS NULL;

        SN RANK       NAME               IB         RB         BS
---------- ---------- ---------- ---------- ---------- ----------
    290000 LIEUTENANT LIM,S.N.                     500        480
    290123 LIEUTENANT UI,C.H.                      450        410
```

6. Ordering Rows of a Query Result.

Q16: List SN, RANK, NAME, IB, and RB of all officers who do not receive IB and in order by their RB.

```
UFI> SELECT PERSON.SN,RANK,NAME,IB,BS,RB FROM PERSON,SALARY
  2  WHERE PERSON.SN = SALARY.SN
  3  AND IB IS NOT NULL
  4  ORDER BY RB;

        SN RANK       NAME               IB         BS         RB
---------- ---------- ---------- ---------- ---------- ----------
    256000 CAPTAIN    CHU,K.S.          960        650        700
    245678 MAJOR      KIM,K.S.          170        820        880
    234567 COLONEL    CHA,S.H.          200        960       1000
    214111 COLONEL    JANG,U.I.         200       1100       1200
    214000 COLONEL    YOON,I.S.         200       1200       1250

5 records selected.
```

B. GROUP FUNCTION

Q17: Find the average basic salary for 'COLONEL'.

```
UFI> SELECT AVG(BS)
  2  FROM SALARY,PERSON
  3  WHERE PERSON.SN = SALARY.SN
  4  AND RANK = 'COLONEL';

   AVG(BS)
----------
1086.66667
```

Q18: List the number of officers whose MOS is 111.

```
UFI> SELECT COUNT(*)
  2  FROM PERSON
  3  WHERE MOS = 111;

  COUNT(*)
----------
         3
```

C. JOINING TWO OR MORE TABLES.

1. Selecting data from two or more tables and equijoin.

Q19: Find SN, RANK, and NAME of all officers from the PERSON table and IB, RB, and BS of all officers from the SALARY table.

```
UFI> SELECT PERSON.SN,RANK,NAME,IB,RB,BS
  2  FROM PERSON,SALARY
  3  WHERE PERSON.SN = SALARY.SN;

        SN RANK       NAME               IB         RB         BS
---------- ---------- ---------- ---------- ---------- ----------
    234567 COLONEL    CHA,S.H.          200       1000        960
    245678 MAJOR      KIM,K.S.          170        880        820
    290000 LIEUTENANT LIM,S.N.                     500        480
    256000 CAPTAIN    CHU,K.S.          960        700        650
    214111 COLONEL    JANG,U.I.         200       1200       1100
    214000 COLONEL    YOON,I.S.         200       1250       1200
    290123 LIEUTENANT UI,C.H.                      450        410

7 records selected.
```

2. Outer-John

Q20: List SN, RANK, NAME and BS of all officers who have received IB and in order by their service number.

```
UFI> SELECT PERSON.SN,RANK,NAME,BS
  2  FROM PERSON,SALARY
  3  WHERE PERSON.SN = SALARY.SN(+)
  4  AND SALARY.IB IS NOT NULL
  5  ORDER BY PERSON.SN;

        SN RANK       NAME               BS
---------- ---------- ---------- ----------
    214000 COLONEL    YOON,I.S.        1200
    214111 COLONEL    JANG,U.I.        1100
    234567 COLONEL    CHA,S.H.          960
   2456078 MAJOR      KIM,K.S.          820
    256000 CAPTAIN    CHU,K.S.          650

5 records selected.
```

D. COMPOUND QUERIES WITH MULTIPLE SUBQUERIES

Q21: List RANK, NAME, MOS, and BS of all officers who have the same MOS as 'CHA, S. H.'.

```
UFI> SELECT RANK,NAME,MOS,BS
  2  FROM PERSON,SALARY
  3  WHERE      PERSON.SN = SALARY.SN
  4  AND        MOS IN
  5             (SELECT MOS
  6              FROM PERSON
  7              WHERE NAME = 'CHA, S. H.');

RANK       NAME              MOS         BS
---------- ---------- ---------- ----------
COLONEL    CHA,S.H.          111        960
LIEUTENANT LIM,S.N.          111        480
CAPTAIN    CHU,K.S.          111        650
```

III. DATA DICTIONARY

A. TABLES THAT DESCRIBE OTHER TABLES.

Q22: List the tables CHA created.

```
UFI> SELECT * FROM TAB;

TNAME                          TABTYPE
------------------------------ -------
AA                             TABLE
CTYPE                          TABLE
FLP                            TABLE
MSTATUS                        TABLE
PERSON                         TABLE
PERSON1                        TABLE
SALARY                         TABLE
TOL                            TABLE

8 records selected.
```

The dictionary table TAB contains the names and descriptions of all the tables, views, synonyms and clusters that user have created. Since user are logged on as user CHA, user see the tables CHA has created

Q23: List all the tables and views that CHA has privileges on.

```
UFI> SELECT * FROM CATALOG;

TNAME                      CREATOR                        TABTYPE      TABID
-------------------------- ------------------------------ ------- ----------
AA                         CHA                            TABLE        26113
CTYPE                      CHA                            TABLE        26625
FLP                        CHA                            TABLE        27137
MSTATUS                    CHA                            TABLE        27649
PERSON                     CHA                            TABLE        28161
PERSON1                    CHA                            TABLE        28673
SALARY                     CHA                            TABLE        29185
TOL                        CHA                            TABLE        29697

8 records selected.
```

The CATALOG list includes the other tables that were created by other user but on which CHA has access privileges. But even the CATALOG list is not complete because it does not contain table from the dictionary.

Q24: List all CHA's tables and view including the dictionary.

```
UFI> SELECT * FROM SYSCATALOG;

TNAME                        CREATOR                      TABTYPE      TABID
---------------------------- ---------------------------- ------- ----------
HELP                         SYSTEM                       TABLE         9985
DUAL                         SYSTEM                       TABLE        10497
STORAGE                      SYSTEM                       VIEW         11520
EXTENTS                      SYSTEM                       VIEW         11776
SPACES                       SYSTEM                       VIEW         12288
SYSCOLUMNS                   SYSTEM                       VIEW         12544
COLUMNS                      SYSTEM                       VIEW         12800
SYSCATALOG                   SYSTEM                       VIEW         13056
CATALOG                      SYSTEM                       VIEW         13312
SYSINDEXES                   SYSTEM                       VIEW         13568
INDEXES                      SYSTEM                       VIEW         13824
VIEWS                        SYSTEM                       VIEW         14080
SYSTABAUTH                   SYSTEM                       VIEW         14336
TAB                          SYSTEM                       VIEW         14848
COL                          SYSTEM                       VIEW         15104
EXPTAB                       SYSTEM                       VIEW         15360
EXPVEW                       SYSTEM                       VIEW         15616
DTAB                         SYSTEM                       TABLE        15873
AA                           CHA                          TABLE        26113
CTYPE                        CHA                          TABLE        26625
FLP                          CHA                          TABLE        27137
MSTATUS                      CHA                          TABLE        27649
PERSON                       CHA                          TABLE        28161
PERSON1                      CHA                          TABLE        28673
SALARY                       CHA                          TABLE        29185
TOL                          CHA                          TABLE        29697

26 records selected.
```

The dictionary table SYSCATALOG includes all tables including the dictionary.

IV. REPORT GENERATION

Q25: Change the heading of each column to two or more lines. Put the title of OFFICER LIST - REGULAR REPORT on separate lines at the top, and CONFIDENTIAL at the bottom of each page of the report.

```
UFI> TTITLE 'OFFICER  LIST||REGULAR REPORT'
UFI> BTITLE 'CONFIDENTIAL'
UFI> COLUMN SN HEADING 'SERVICE|NUMBER'
UFI> COLUMN MOS HEADING 'MILITARY|OCCUPATIONAL|SPECIALTY'
UFI> COLUMN CD HEADING 'COMMISSION|DATE'
UFI> COLUMN CT HEADING 'COMMISSION|TYPE'
UFI> COLUMN MS HEADING 'MARRIAGE|STATUS'
UFI> SELECT      *
  2  FROM        PERSON
  3  ORDER  BY   SN;
```

```
Tue Sep 13                                                                  page    1
                                   OFFICER  LIST

                                   REGULAR REPORT

                                       MILITARY
 SERVICE                           OCCUPATIONAL COMMISSION COMMISSION   MARRIAGE
 NUMBER RANK        NAME              SPECIALTY DATE             TYPE     STATUS
-------- ---------- ---------- ------------ ---------- ---------- ---------
  214000 COLONEL    YOON,I.S.           121 17-DEC-63           2          1
  214111 COLONEL    JANG,U.I.           130 24-JUN-64           1          1
  234567 COLONEL    CHA,S.H.            111 11-MAR-65           1          1
  245001 MAJOR      PARK,S.U.           130 21-SEP-66           1          2
  245678 MAJOR      KIM,K.S.            121 20-APR-66           1          1
  256000 CAPTAIN    CHU,K.S.            111 11-OCT-68           2          1
  290000 LIEUTENANT LIM,S.N.            111 16-JAN-70           1          2
  290123 LIEUTENANT UI,C.H.             222 05-SEP-71           2          2




                                    CONFIDENTIAL

8 records selected.
```

Q26: Put the title of MONTHLY - PERSONNEL REPORT on separate lines at the top of each page and list service number, rank, intelligence benefit, regular bonus, and basic salary of all officers.

```
UFI> TTITLE 'MOMTHLY SALARY||PERSONNEL REPORT'
UFI> COLUMN SSTEP HEADING 'SALARY|STEP'
UFI> COLUMN IB HEADING 'INTELLIGENCE|BENEFIT'
UFI> COLUMN RB HEADING 'REGULAR|BONUS'
UFI> COLUMN BS HEADING 'BASIC|SALARY'
UFI> SELECT PERSON.SN, RANK, NAME, IB, RB, BS
  2  FROM   PERSON, SALARY
  3  WHERE  PERSON.SN = SALARY.SN;
```

```
Tue Sep 13                                                          page    1
                                MOMTHLY SALARY

                               PERSONNEL REPORT

  SERVICE                                INTELLIGENCE    REGULAR      BASIC
   NUMBER RANK        NAME                    BENEFIT      BONUS     SALARY
---------- ---------- ---------- ------------ ---------- ----------
   234567 COLONEL     CHA,S.H.                    200       1000        960
   245678 MAJOR       KIM,K.S.                    170        880        820
   245001 MAJOR       PARK,S.U.                   170        800        710
   290000 LIEUTENANT  LIM,S.N.                               500        480
   256000 CAPTAIN     CHU,K.S.                    960        700        650
   214111 COLONEL     JANG,U.I.                   200       1200       1100
   214000 COLONEL     YOON,I.S.                   200       1250       1200
   290123 LIEUTENANT  UI,C.H.                                450        410

8 records selected.
```

LIST OF REFERENCES

1. James, A. O'Brien , Computer in Business Management, Third Edition, 1982, Richard D. Irwin, Inc.

2. Martin, J., Computer Database Organization, Prentice-Hall Inc., Englewood Cliffs, NJ, 07632, 1977.

3. Kroenke, D., Database Processing, Science Research Associate, Inc., Chicago, Tronto, 1977.

4. Date, C. J., An Introduction in Database System, Third Edition, 1981, Addision-Wesley Pub. Co.

5. Codd, E. F.(1970), "A Relational Model of Data for Large Shared Data Banks," Commun. ACM 13, pp. 337-387.

6. Chamberlin, D. D.(1976), "Relation Database Management System," ACM Comput. Surv. 8, pp. 43-66.

7. Codd, E. F.(1972a), "Further Normalization of the Database Relational Model," in Database Systems Courant Comput. Sci. Symp. 6th (Rustin, R., ed.), pp. 65-98, Prentice-Hall, Englewood Cliffs, NJ.

8. Code, E. F.(1972b), "Relational completeness of database sublanguages," Ibid., pp. 65-98.

9. Korea Air Force Unit Education and Training Regulation, HQ of Air Force.

10. Korea Air Force Personnel Management Regulations, HQ of Air Force.

11. Tsichritzis, D. C. and Lochovsky, F. H., Data Models, Prentice-Hall Inc., Englewood Cliffs, NJ, 07632, 1982.

12. Weldon, J. L.(1981), Data Base Administration, Plenum Press, New York.

13. Tsichritzis, D. C., and Klug, A.(eds.)(1978), "The ANSI/X3/SPARC DBMS Framework Report of the Group and Database Management Systems," Inf. Syst. 3, pp. 173-191.

14. BCS(1977), "The British Computer Society Data Dictionary Systems Working Party Record." ACM SIGMOD Record 9(4), pp. 2-24.

15. The Design of an Integrated Data Dictionary System, Requirements and logical Structures Preceedings, New York, May 1978, Edited by S. B. Yao, S. B. Navathe, J. L. Weldon, and T. L. Kunil, pp. 56-71.

16. Ullman, J. D., Principle of Database System, Computer Science Press, Inc., Rookville, Maryland, 20850, 1980.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Professor David K. Hsiao, Code 52Hq Department Chairman Department of Computer Science Naval Postgraduate School Monterey, California 93943	2
4.	Assistant Professor Daniel R. Dolk, Code 54Dk Department of Administrative Science Naval Postgraduate School Monterey, California 93943	1
5.	Department of Computer Science, Code 52 Naval Postgraduate School Monterey, California 93943	4
6.	Computer Center, Department of The AIR FORCE The Republic of KOREA AIR FORCE SEOUL, KOREA	4
7.	LTC Lim, S. Nam SMC 2852 Naval Postgraduate School Monterey, California 93943	1
8.	Cha, Sang Ho #206-40 Hong_Eun_3_Dong Seodaemun_gu SEOUL 122, KOREA	6
9.	Major S. K. Kim SMC 2793 Naval Postgraduate School Monterey, California 93943	1

	No. Copies
10. Curricular Officer, Code 37 Computer Technology Programs Naval Postgraduate School Monterey, California 93943	1